



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

EFFECTIVIDAD DE LOS MODELOS JERÁRQUICOS
Y TRADICIONALES PARA EL RECONOCIMIENTO
DEL HABLA EN EL LENGUAJE ESPAÑOL DE
MÉXICO.

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

Ingeniero en Computación

PRESENTA:

David Andrés Vásquez Anguiano

TUTOR:

M. C. Carlos Oliver Morales



Aragón, Edo. Mex., 2022



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Dedicado a mi familia, gracias por haberme brindado su apoyo incondicional además de ser el motor y razón por la cual me esfuerzo en cumplir mis sueños y metas.

Es a ustedes que dedico el presente proyecto de tesis.

Agradecimientos

Existe en Japón un dicho que traducido al español dice más o menos:

”La disciplina tarde o temprano vencerá a la inteligencia”

Esto quiere decir que el éxito será de quienes se esfuerzan de forma perseverante y organizada y no de los que tienen un talento nato sin desarrollar. Con dicha filosofía en mente y siguiendo los consejos de las siguientes personas es que se llegó a la finalización de esta tesis.

Para comenzar me gustaría agradecer especialmente a mi padre Rafael Vásquez Mendoza, mi madre Antonia Anguiano Gonzáles y mis hermanos Erick Vásquez Anguiano y Angel Vásquez Anguiano. Quiénes me han ayudado a superar todos los obstáculos presentados y me han animado a persistir hasta la culminación de este trabajo académico.

De igual modo quiero agradecer a la familia Vázquez Mendoza y Anguiano Gonzáles por todo el soporte dado durante mi vida universitaria.

Asimismo, quiero mostrar mi agradecimiento a mi director de tesis el M. C. Carlos Oliver Morales por brindarme su apoyo y compartir conmigo sus valiosos conocimientos y tiempo.

Por último, expreso mi inmensa gratitud a mi alma mater, la Universidad Nacional Autónoma de México por enriquecerme en conocimiento y brindarme las oportunidades de crecimiento personal y profesional.

Resumen

El término de Clasificación Tradicional es utilizado para definir la etapa en que los algoritmos de clasificación asociaban un objeto con una única clase. Sin embargo, algunos problemas de clasificación no siempre siguen este comportamiento. Dicho inconveniente dio origen al término de Clasificación Jerárquica, quien representa un nuevo tipo de algoritmo de clasificación en donde el objetivo principal es dividir los objetos a clasificar en subclases siguiendo una taxonomía pre-establecida o generada automáticamente para después, haciendo uso de la taxonomía, clasificar cada objeto con más de una clase. La finalidad de esta tesis es comprobar que los algoritmos de clasificación jerárquica presentan un mejor rendimiento en la tarea de clasificación de los 22 alófonos principales que conforman al lenguaje Español hablado en México, comparado con los algoritmos de clasificación tradicionales como lo son las Máquinas de Vectores de Soporte y el Perceptrón Multicapa. Se optó por crear representaciones MFCC de cada objeto presente en nuestro Corpus de datos con el objetivo de contar con una representación practica de los mismos. Resultando subsecuentemente en la utilización de programación paralela y archivos llamados Metadatas buscando de esa forma disminuir el tiempo de ejecución de cada programa. Por otro lado, el Modelo Jerárquico evaluado pertenece al enfoque Local por Nodo Padre, el cual asimismo implementa una política de entrenamiento Siblings, clasificadores binarios (siendo estos perceptrones multicapa) y una taxonomía preestablecida. Además se forzó de forma intencional una sobrecarga de datos en cada Nodo Padre con la finalidad de que cada clasificador se encargue únicamente de decidir si el objeto en turno pertenece o no a su clase encargada, repitiendo el mismo proceso hasta terminar en un Nodo Hoja. Las eficiencias medidas por los Modelos Tradicionales y Jerárquicos fueron obtenidas mediante la métrica F1, siendo nuestra eficacia final el promedio calculado de 30 ejecuciones correspondientes a los Modelos Tradicionales y Jerárquico obteniendo de esa forma una comprobación estadística, concluyendo en que el Modelo Jerárquico evaluado en esta tesis presenta una eficacia promedio mayor a la obtenida por los Modelos SVM y MLP aunque con un tiempo de ejecución y complejidad mucho mayor.

Abstract

The term Traditional Classification is used to define the stage in which classification algorithms associate an object with a single class. However, some classification problems do not always follow this behavior. This inconvenience gave rise to the term Hierarchical Classification, which represents a new type of classification algorithm where the main objective is to divide the objects to be classified into subclasses following a pre-established or automatically generated taxonomy for later, making use of this taxonomy, classify each object with more than one class. This thesis aims to verify that hierarchical classification algorithms present a better performance in the classification task of the 22 main allophones present in the Spanish language spoken in Mexico, compared to traditional classification algorithms such as Support Vector Machines Multi-layer Perceptron. It was decided to create MFCC representations of each object present in our data Corpus to have an efficient and practical representation of them. Subsequently results in the use of parallel programming and files called Metadata, thus seeking to reduce the execution time of each program. On the other hand, the evaluated Hierarchical Model belongs to the Parent Node Local approach, which also implements a Siblings training policy, binary classifiers (these being multilayer perceptrons), and a pre-established taxonomy. In addition, an overload of data was intentionally forced in each Parent Node so that each classifier is solely responsible for deciding whether or not the object in turn belongs to its assigned class, repeating the same process until ending up in a Leaf Node. The efficiencies measured by the Traditional and Hierarchical Models were obtained using the F1 metric, with our final efficiency being the calculated average of 30 runs corresponding to the Traditional and Hierarchical Models, thus obtaining a statistical verification, concluding that the Hierarchical Model evaluated in this thesis presents an average efficiency greater than that obtained by the SVM and MLP Models, although with a much greater execution time and complexity.

Índice general

Resumen	V
Abstract	VII
Índice de figuras	XI
Índice de tablas	XIII
Introducción	XV
1. Terminología	1
1.1. Clasificación	1
1.2. Dataset	2
1.3. Corpus	2
1.4. Fonema	2
1.5. Alófono	2
1.6. Diferencia entre Fonema y Alófono	2
1.7. Modelo de Clasificación	3
1.8. Coeficientes Cepstrales en las Frecuencias de Mel (MFCC)	3
2. Clasificación Tradicional	5
2.1. Clasificación Supervisada	5
2.2. Algoritmos de Clasificación	6
2.2.1. Perceptrón Multicapa (MLP)	6
2.2.2. Máquinas de Vectores de Soporte (SVM)	8
3. Clasificación Jerárquica	11
3.1. Métodos de Clasificación Jerárquica	12
3.2. Clasificación Jerárquica Plana	12
3.3. Clasificación Jerárquica Local	12
3.3.1. Clasificación Jerárquica Local por Nodo	13
3.3.2. Clasificación Jerárquica Local por Nodo Padre	15
3.3.3. Clasificación Jerárquica Local por Nivel	16

4. Comparación Modelos Tradicionales y Jerárquicos	17
4.1. Antecedentes Históricos	18
4.2. Medidas de Evaluación	19
4.3. Analisis de Varianza (ANOVA)	20
4.4. Método de Turkey	22
5. Construcción de los Modelos Tradicionales y Jerárquico	23
5.1. Preparación del DataSet	24
5.2. Preprocesamiento de los datos	24
5.3. Creación y Ajuste de los Modelos Tradicionales	25
5.4. Creación y Ajuste del Modelo Jerárquico	26
6. Obtención de Resultados	29
6.1. Experimentación	29
6.2. Precisión en el Reconocimiento de Alófonos de los Modelos Tradicionales	29
6.3. Precisión en el Reconocimiento de Alófonos de un Modelo Jerárquico . .	31
6.4. Analisis de Varianza para los Modelos Tradicionales y Jerárquico	31
6.5. Aplicación del Método de Tukey	33
7. Conclusión	35
7.1. Resumen	35
7.2. Conclusiones	36
7.3. Aportaciones y Contribuciones	37
7.4. Trabajo a Futuro	37
Bibliografía	39
A. Lista de Variables	43
B. Alófonos Nivel T22	45
C. Bitácora del DataSet	47

Índice de figuras

1.1. Ejemplo de Clasificación.	1
1.2. Representacion de una Onda de Audio en MFCC.	3
2.1. MLP con una Capa Oculta.	7
2.2. Ejemplo del Hiperplano de un SVM.	9
3.1. Clasificación Jerárquica Plana.	13
3.2. Clasificación Jerárquica Local por Nodo.	14
3.3. Clasificación Jerárquica Local por Nodo Padre.	15
3.4. Clasificación Jerárquica Local por Nivel.	16
4.1. Tabla de Cuantiles de la Distribución de Tukey.	22
5.1. Funcionamiento General de los Modelos de Clasificación.	25
5.2. Taxonomía del Modelo Jerárquico.	26

Índice de tablas

4.1. Características de los Modelos Tradicionales y Jerárquicos.	17
4.2. Ejemplo de una Matriz de Confusión.	19
4.3. Tabla para el Análisis de Varianza.	21
6.1. Precisión en el Reconocimiento de Alófonos de un MLP.	30
6.2. Precisión en el Reconocimiento de Alófonos de un SVM.	30
6.3. Precisión en el Reconocimiento de Alófonos de un Modelo Jerárquico. . .	31
6.4. Tabla para el Análisis de Varianza de los Modelos Tradicionales y Jerárquico.	32
6.5. Tabla para el Análisis de Varianza de los Modelos Tradicionales y Jerárquico.	33
B.1. Consonantes del Nivel T22 en el Corpus DIMEx100.	45
B.2. Vocales del Nivel T22 en el Corpus DIMEx100.	45

Introducción

El idioma Español es uno de los lenguajes más hablados en el mundo, esto debido a que se encuentra presente como idioma oficial en casi todos los países del continente Americano por lo que subsecuentemente se convierte en uno de los lenguajes más importantes en el mundo. Es por esta razón que se escogió al idioma Español, específicamente al hablado en México como la base de nuestra investigación. Para fines de esta tesis se tomara a la efectividad como el sinónimo de eficacia, sustituyendo a la misma a lo largo de este documento.

Además del hecho de considerarlo más accesible para realizar esta investigación y por el deseo de aportar mediante este trabajo al escaso campo del reconocimiento del habla en dicho idioma. A pesar de las razones antes mencionadas la importancia de esta lengua queda opacada por el poco o nulo interés en la misma. Siendo superada por el idioma Inglés recibiendo la mayor atención en el área de la investigación y por parte del sector privado.

Este documento busca cumplir con un objetivo principal, el cual es el evaluar la eficacia obtenida en el reconocimiento de los 22 alófonos principales que conforman al Español Mexicano clasificados mediante la implementación de Modelos Tradicionales y Jerárquicos con la finalidad de conseguir la mejor opción disponible para la creación de reconocedores del habla en este idioma.

Asimismo se cuenta con objetivos secundarios los cuales se presentan a continuación:

1. Diseñar y ajustar los parámetros del modelo “Máquinas de Vectores de Soporte“ buscando obtener la mayor precisión posible en el menor tiempo.
2. Diseñar y ajustar los parámetros del modelo “Perceptrón Multicapa“ en donde se consiga la mayor precisión posible en el menor tiempo.
3. Diseñar e implementar el “Modelo Jerárquico“ buscando obtener la mejor precisión posible en el menor tiempo.

Cabe resaltar que la hipótesis del presente documento de tesis se basa en que los Modelos Jerárquicos ofrecen una eficacia superior comparada con los Modelos Tradicionales en el ámbito del reconocimiento de los 22 alófonos principales pertenecientes al Español Mexicano aplicando para ello una demostración estadística y la métrica F1.

El presente documento de investigación está conformado por siete capítulos a través de los que se abordarán todos los temas involucrados en este trabajo de tesis. Iniciando con el primer capítulo, “Terminología“ en donde se definen conceptos que serán retomados a lo largo de esta tesis y que juegan un papel importante para el entendimiento de la misma.

Siguiendo con el segundo capítulo y tercer capítulo, “Clasificación Tradicional“ y “Clasificación Jerárquica“ correspondientemente se definen de manera formal los clasificadores planos y jerárquicos a evaluar junto con la explicación de los mismos.

En el cuarto capítulo, “Comparación Modelos Tradicionales y Jerárquicos“ se presentan los antecedentes históricos de investigaciones anteriores cuya finalidad fueron la comparación, búsqueda de mejores rendimientos y aplicación práctica de los modelos construidos en esta tesis junto con las métricas de evaluación que serán utilizadas para medir las eficacias de nuestros modelos.

Posteriormente en el quinto capítulo, “Construcción de los Modelos Tradicionales y Jerárquicos“ se adjuntan todos los pasos tomados en la creación y ajuste de los modelos evaluados junto con el tratamiento dado a los datos utilizados. En el sexto capítulo, “Obtención de Resultados“ se presentan las eficacias resultantes de la experimentación en la clasificación de los 22 alófonos del idioma Español.

Resultando finalmente en el séptimo capítulo “Conclusión“, en donde como su nombre lo indica se presentan las conclusiones a las cuales se llegaron una vez finalizada esta investigación, las aportaciones y contribuciones al igual que el trabajo futuro derivado de la misma.

Terminología

A lo largo de este capítulo se presentaran los conceptos que serán regularmente mencionados a lo largo de la presente tesis y que son de interés si se desea lograr un mejor entendimiento de la misma.

Cabe aclarar que algunos términos no tendrán una relación directa entre ellos y que no serán abordados a profundidad debido a no ser el tema central de esta investigación.

1.1. Clasificación

Actividad de agrupar un conjunto de objetos en un número de clases ya sean predefinidas o encontradas de forma automática mediante el uso de técnicas de inteligencia artificial o estadísticas, considerando que cada objeto cuenta con un conjunto de características propias o compartidas [Hernández, 2012].

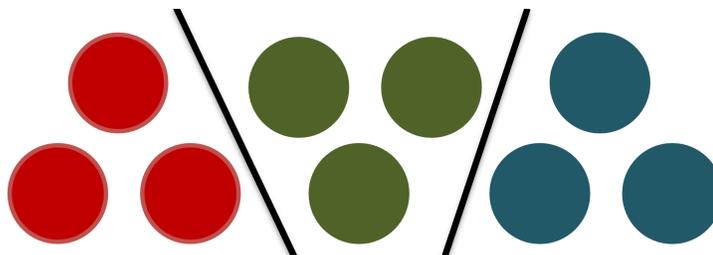


Figura 1.1: Ejemplo de Clasificación.

[Vásquez, 2022]

1.2. Dataset

Conjunto de datos correspondientes al fenómeno que deseamos clasificar, generalmente una tabla de formato simple [Provost and Kohavi, 1998].

1.3. Corpus

Recopilación de textos naturales (no creados expresamente para su incorporación en el corpus), almacenados de forma electrónica, representativos de una variedad lingüística, reunidos con el propósito de facilitar su estudio científico [Rojo, 2016].

1.4. Fonema

Unidad fonológica abstracta más pequeña con la que podemos distinguir un sonido y/o palabra. Por ejemplo, el fonema /a/ en la palabra “mamá”. [Rebollo, 1981].

1.5. Alófono

Representación del sonido de un fonema de acuerdo a la posición en la que este se encuentre. Un ejemplo sería el fonema /d/ en la palabra “dado” se relaciona los alófonos, [d] y [ð] [Diaz and Ramirez, 2018].

1.6. Diferencia entre Fonema y Alófono

La principal diferencia entre los fonemas y alofonos recae en que los alofonos representan las pronunciaciones reales de un fonema sin afectar al significado de una palabra.

Para entender la relación entre fonemas y alófonos, hay que saber que un fonema no se produce siempre de la misma manera. Este depende de factores como otros sonidos en contacto, el estilo de habla, la rapidez de elocución, etc.

Resultando en que se puedan realizar transcripciones en dos tipos principales, la fonémica /a/ y la alofónica [a]. Por ejemplo, la palabra inglesa night (noche en español) puede ser transcrita como /nixt/ de forma fonémica.

Por otro lado, se puede observar que el fonema /x/ se pronuncia [ai] cuando hay presencia de alargamiento en el fonema /i/, siendo [náit] la transcripción alofónica.

Las transcripciones fonémicas suelen ser bastante anchas y por lo general poco detalladas. Mientras que las transcripciones alofónicas pueden llegar a ser muy detalladas y pequeñas, variando según las necesidades [Álvarez, 2020].

1.7. Modelo de Clasificación

A veces llamado simplemente *Modelo*. Es el resultante de resumir total o parcialmente un dataset por medio de un algoritmo de clasificación para después este mismo ser utilizado para descripción o predicción [Provost and Kohavi, 1998].

1.8. Coeficientes Cepstrales en las Frecuencias de Mel (MFCC)

Se basan en el sistema auditivo humano. La percepción humana de los contenidos de frecuencia de los sonidos para las señales del habla no sigue una escala lineal. Por lo tanto, por cada tono con una frecuencia medida en Hz le corresponde un tono equivalente medido en una escala llamada *Mel Scale*. El MFCC tiene la habilidad de representar los espectros de amplitud del habla en una forma compacta [Tiwari, 2010].

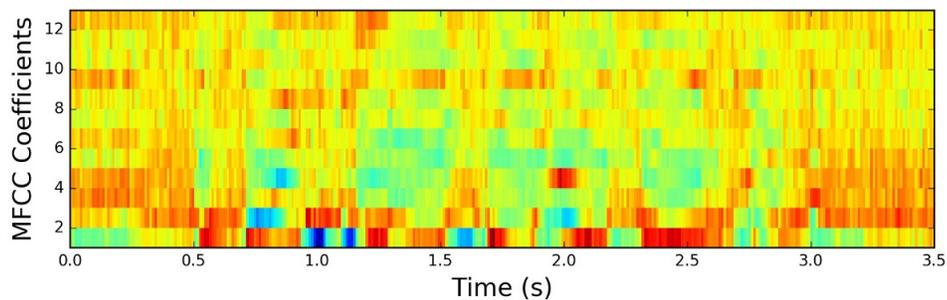


Figura 1.2: Representación de una Onda de Audio en MFCC.

[Pratheeksha, 2018]

Clasificación Tradicional

Debido a la necesidad de clasificar de forma automática y eficiente se desarrollaron procesos de clasificación, los que a su vez dieron origen a algoritmos para llevar a cabo la tarea de clasificación. A esta etapa en los algoritmos de clasificación se le conoce como **clasificación tradicional**, clasificación en donde se asocia cada objeto x con una sola clase l de un conjunto finito de clases L , lo que en otras palabras indica que un objeto pertenece únicamente a una clase [Hernández, 2012].

Los **procesos de clasificación** dieron origen a su vez a dos paradigmas de aprendizaje: **clasificación supervisada** y **clasificación no supervisada**. La clasificación supervisada crea un modelo de clasificación basado en un entrenamiento a partir de un conjunto de objetos preclasificados y haciendo uso del mismo se clasifican los objetos restantes o de prueba. La **clasificación no supervisada** realiza agrupamientos de objetos tomando como referencia las similitudes encontradas entre los mismos [Haykin and Network, 2004].

Las siguientes secciones abordan los temas relacionados con la clasificación supervisada al igual que los algoritmos de clasificación utilizados en la presente tesis. Véase el apéndice A para consultar la lista de variables.

2.1. Clasificación Supervisada

La **clasificación supervisada** recibe su nombre gracias a que un experto humano es el encargado de “supervisar” que el modelo se comporta como es esperado o en caso contrario ajustar los parámetros necesarios. Una pieza clave dentro de la clasificación supervisada es el **dataset de entrenamiento**, definido por un vector de características y la clase perteneciente del objeto. Por último, haciendo uso del **dataset de prueba** y de alguna métrica de evaluación se medirá el correcto funcionamiento del mismo [Haykin and Network, 2004].

La clasificación supervisada puede dividirse a su vez en dos subclases: **clasificación binaria** la cual tiene como objetivo el determinar si un objeto pertenece o no a una clase, generalmente usada en problemas de clasificación multidimensional y **clasificación multi-clase** que como su nombre lo indica discierne entre dos o más clases siendo por lo tanto la más común [Hernández, 2012].

Debemos tener cuidado cuando entrenamos nuestro modelo debido a que pueden presentarse casos de sobreajuste o desajuste. El **sobreajuste** aparece cuando nuestro modelo durante la fase de entrenamiento no mejoró su habilidad de clasificar dando como resultado un mal rendimiento al momento de clasificar datos desconocidos. Contrario al sobreajuste tenemos al **desajuste**, donde nuestro modelo no fue capaz de captar el patrón necesario dentro de nuestros datos de entrenamiento [Koehrsen, 2018].

2.2. Algoritmos de Clasificación

Los algoritmos de clasificación son herramientas utilizadas para establecer una relación entre una clase y un objeto determinado de acuerdo a las características propias del objeto a clasificar creando con ello un modelo de clasificación. En la siguiente sección se profundiza en los algoritmos de clasificación utilizados en la presente tesis para la construcción de nuestros modelos tradicionales.

2.2.1. Perceptrón Multicapa (MLP)

Para entender lo que es un perceptrón multicapa debemos inicialmente comprender lo que es un perceptrón o neurona. El **perceptrón** tiene el objetivo de inferir la clase de un objeto de entre dos clases posibles por ejemplo 0 y 1 en el caso de un clasificador binario. Subsecuentemente, podemos decir que un **perceptrón multicapa** (MLP) no es más que un set de perceptrones agrupados en capas conectadas total o localmente más comúnmente conocido como red neuronal [Van, 1986].

A continuación se enlistan algunos de los elementos primordiales que podemos encontrar dentro de un perceptrón multicapa recopilados de [Haykin and Network, 2004]:

1. Un set de **sinapsis** o **links de conexión**, cada uno con un peso propio.
2. Una **función de activación no lineal** $\varphi(\cdot)$, encargada de limitar el rango de amplitud de una señal de salida a un valor finito, por ejemplo $[0,1]$. Algunos ejemplos de funciones no lineales son la tangente hiperbólica [Abramowitz and Stegun, 1972] y la función rectificadora (ReLU) [Hahnloser et al., 2000].

3. Un **bias** que incrementa o decrementa la entrada de una red de una función de activación.
4. Un **learning rate** (ganancia), medida en que nuestro modelo se ajusta de acuerdo a los datos que se le presentan.

De acuerdo con [Pedregosa et al., 2013], un perceptrón multicapa debe su aprendizaje a una función $f(\cdot) : R^i \rightarrow R^o$ aplicada sobre un dataset de entrenamiento, donde i es la cantidad de neuronas de entrada y o es la cantidad de neuronas de salida. El MLP necesita de un set de features $X = x_1, x_2, \dots, x_m$ los cuales representan las características de un objeto y un target y , equivalente a la clase del objeto.

En la figura (2.1) se muestra un ejemplo de un MLP donde las neuronas se encuentran conectadas totalmente y se cuenta con una capa oculta a , features X , un bias de +1 y un escalar de salida $f(X)$. La primera capa (**capa de entrada**), consiste en un set de neuronas $\{x_i | x_1, x_2, \dots, x_m\}$ representando cada una los features de entrada. Posteriormente se encuentran las neuronas de la **capa oculta** (“oculta” porque no pertenece ni a la capa de entrada o salida) las que se encargan de transformar los valores de la capa previa con una suma lineal de pesos $w_1x_1 + w_2x_2 \dots w_nx_n$, seguido por una función de activación no lineal. Al final nos encontramos con la **capa de salida** recibe los valores de la última capa oculta y los transforma a valores de salida.

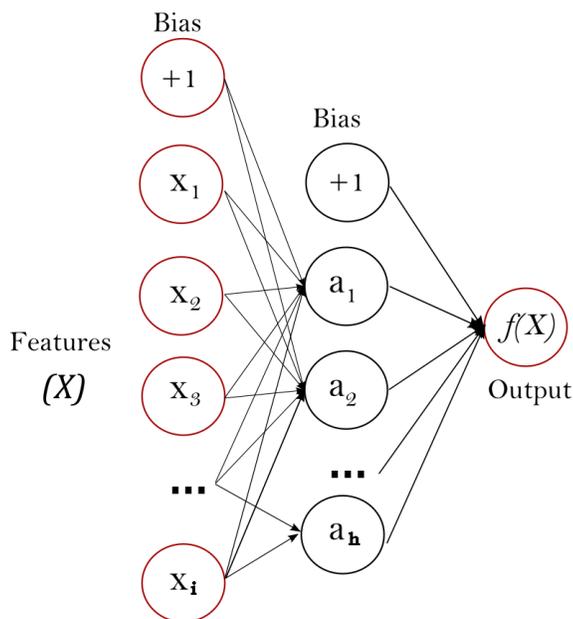


Figura 2.1: MLP con una Capa Oculta.

[Pedregosa et al., 2011]

2. CLASIFICACIÓN TRADICIONAL

El MLP se apoya en parámetros para ajustar el modelo buscando así mejorar el rendimiento de nuestro algoritmo. Dentro de estos podemos destacar a **Alpha** y **Solver**. Alpha es un parámetro usado para combatir el sobreajuste. Incrementar el alpha podría arreglar el sobreajuste ya que se da más relevancia a los pesos más pequeños, resultando a su vez en un **diagrama de límite de decisión** menos complejo. Por otra parte, si decrementamos el alpha podríamos combatir el desajuste dando más relevancia a los pesos más grandes resultando así en un diagrama más complejo.

Adicionalmente, el algoritmo MLP se entrena haciendo uso de algoritmos de optimización llamados solvers, estos nos ayudan en general a obtener el mejor rendimiento de nuestro modelo dependiendo de cómo se presenten nuestros datos. Dentro de los solvers que podemos encontrar en un MLP está el descenso de gradiente estocástico (SGD), Adam y L-BFGS. Observando empíricamente encontramos que L-BFGS tiene un tiempo de complejidad menor y con mejores resultados en datasets pequeños. Por otro lado, Adam se comporta de mejor manera con grandes datasets. No obstante, SGD puede llegar a presentar un rendimiento aún mejor que los anteriores en el caso de que la ganancia se ajustó correctamente [Pedregosa et al., 2013].

En conclusión, las ventajas de los perceptrones multicapa son su capacidad de aprender en tiempo real. Mientras que dentro de sus desventajas encontramos que se requiere ajustar un gran número de parámetros volviéndolo tedioso y tardado. Además de ser sensible a los datos escalados produciendo resultados distintos dependiendo si están o no los datos previamente procesados.

2.2.2. Máquinas de Vectores de Soporte (SVM)

Las **máquinas de vectores de soporte** (SVM) trabajan bajo la idea de construir hiperplanos que sirvan como la base de separación entre los datos para que mediante ellos se realice la acción de clasificar. Para definir el margen e hiperplano óptimo se recurre a los **vectores de soporte** los que a su vez son datos mapeados en el espacio que cumplen la característica de tener entre ellos la mayor distancia posible en el espacio. Por último, se mapean los datos de prueba obteniendo de esta forma la clase a la que pertenece el objeto a clasificar de acuerdo al espacio donde este sea ubicado [Haykin and Network, 2004].

En la figura (2.2) podemos observar el hiperplano de una SVM con un espacio de 2 dimensiones. Adicionalmente nos encontramos con los vectores de soporte marcados con cuadros grises, quienes se encuentran separando las dos clases (cruces y círculos). Se puede notar igualmente la aparición de un **hiperplano óptimo** definido como la función lineal con máximo margen entre vectores de dos clases implementado para la separación de datos de acuerdo a su clase correspondiente.

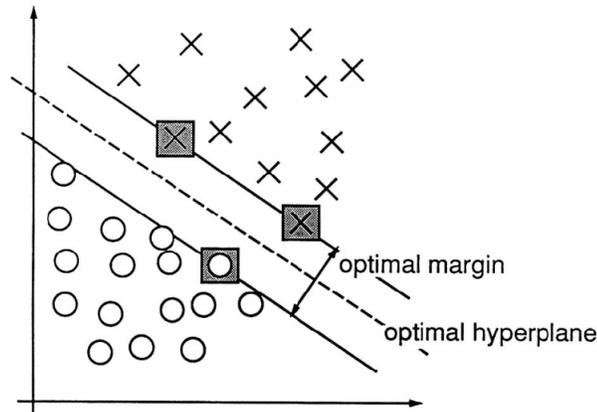


Figura 2.2: Ejemplo del Hiperplano de un SVM.

[Cortes and Vapnik, 1995]

Debido a que los objetos que buscamos clasificar cuentan por lo general con una gran cantidad de atributos y/o clases. Se recurre entonces al uso de un **kernel** para transformar los datos a un espacio de alta dimensión con un bajo costo computacional y alta eficacia. Dentro de las funciones de kernel más comunes tenemos el lineal, polinomial, rbf y sigmoid. Cuando se entrena una SVM con un kernel del tipo **Radial Basis Function (rbf)** se deben considerar los parámetros C y γ . El parámetro C busca compensar la clasificación errónea durante el entrenamiento, en cambio el parámetro **gamma** (γ) define que tanta influencia tiene un ejemplo sobre el otro durante el entrenamiento.

Las SVMs son poderosas herramientas. No obstante, tienden a convertirse en modelos robustos incrementando el tiempo de cómputo considerablemente en caso de tener un dataset de entrenamiento muy grande. Al hacer uso de un kernel la complejidad en la tarea de entrenamiento se complica dependiendo del número de parámetros a optimizar por lo que es recomendable iniciar el entrenamiento de una SVM con el kernel rbf ya que este cuenta solo con dos parámetros C y γ , los que a su vez se comparten con los demás kernels.

Finalmente, las ventajas de usar máquinas de vectores de soporte son su eficiencia en espacios de altas dimensiones, eficacia en casos donde el número de dimensiones sea mayor al número de ejemplos, eficiencia de memoria debido al uso de vectores de soporte y versatilidad ya que se puede combinar con distintos kernels de acuerdo a nuestras necesidades. Sin embargo, algunas de las desventajas son que si el número de features es mucho más grande que el número de ejemplos, evitar el sobreajuste mediante elegir la función de kernel y los términos de regularización correctos se vuelve una tarea crucial [Pedregosa et al., 2013].

Clasificación Jerárquica

La **Clasificación Jerárquica** también conocida como clasificación estructurada (*Structure Classification*) es un tipo de clasificación en donde se requiere de una taxonomía para llevar a cabo el proceso de clasificación. Este tipo de clasificación se basa en dividir los objetos a clasificar en pequeños subconjuntos para una más óptima clasificación en comparación con la clasificación tradicional.

Aunque al igual que en la clasificación tradicional los métodos de clasificación jerárquica se dividen en supervisados y no supervisados. En el primer grupo las clases son definidas de acuerdo a una taxonomía pre-establecida. Por otro lado, en el segundo grupo no se hace uso de ninguna taxonomía debido a que la misma se genera durante el proceso de entrenamiento y se agrupan las clases similares unas con otras en meta-clases. La taxonomía resultante será la utilizada para el proceso de clasificación [Hernández, 2012].

Se entiende como **Taxonomía** a un árbol o un GAD definido sobre un set (C_T, \prec) donde C_T es un conjunto finito que enumera todas las clases en el dominio de aplicación y \prec representa una relación anti-reflexiva (un nodo se relaciona solamente con nodos de clases inferiores) y transitiva (si tenemos un nodo padre entonces dicho nodo se relaciona con sus nodos hijos, los cuales a su vez se relacionarán con sus propios nodos hijos). La raíz del árbol de una taxonomía es la etiqueta (clase) más general aplicable a cada objeto. Las hojas del árbol indican las etiquetas más específicas. Sin embargo, cuando una etiqueta es asignada a un objeto dentro de la taxonomía, implícitamente es etiquetado con todas las etiquetas pertenecientes a sus ancestros [Wu et al., 2005].

A continuación se presentará un panorama general sobre las diferentes técnicas de clasificación jerárquica existentes. Véase el apéndice A para consultar la lista de variables.

3.1. Métodos de Clasificación Jerárquica

De acuerdo con [Freitas and de Carvalho, 2007] los métodos de clasificación jerárquica cuentan con tres características principales:

1. **Estructura de la jerarquía de clases.** Se trata de la taxonomía en la que se basará el clasificador. Por lo general un árbol o un grafo acíclico dirigido (GAD).
2. **Nivel jerárquico de clases predichas.** Indica si el clasificador debe detener su ejecución una vez alcanzado el nivel de hojas o en cualquier nodo de la jerarquía sin importar que sea nodo hoja o interno, exceptuando al nodo raíz.
3. **Enfoque de clasificación jerárquica.** Hace referencia a la forma en que se recorrerá la taxonomía del clasificador. En este caso nos centraremos en la Clasificación Jerárquica Plana y Clasificación Jerárquica Local.

3.2. Clasificación Jerárquica Plana

El enfoque de **Clasificación Jerárquica Plana** se comporta de la misma forma en que lo harían los modelos tradicionales, debido a que ignora por completo la jerarquía durante los procesos de entrenamiento y clasificación. Basa su funcionamiento en utilizar algoritmos de clasificación multi-clase para predecir los nodos hoja de la jerarquía ignorando la información contenida en las relaciones de los nodos en la jerarquía, lo que se convierte en su principal desventaja.

Por otro lado, tiene la ventaja de contar con una implementación muy simple debido a hacer uso de un solo clasificador multi-clase [Hernández, 2012].

Se representa el enfoque de Clasificación Jerárquica Plana en la figura (3.1). Los círculos representan las clases de la jerarquía mientras que el recuadro punteado representa al clasificador multi-clase. La predicción del clasificador consiste en un solo nodo de color rojo en donde se asume la pertenencia con todos sus ancestros.

3.3. Clasificación Jerárquica Local

La primera mención acerca del método de **Clasificación Jerárquica Local** o *Top-Down* ocurre en el trabajo de [Koller and Sahami, 1997]. En este enfoque se entrena cada clasificador, ya sea un clasificador multi-clase o binario, basándose en la información local de cada nodo en la jerarquía. Durante su fase de entrenamiento se comporta al igual que un clasificador tradicional.

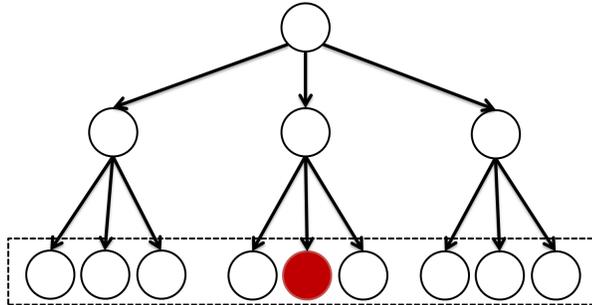


Figura 3.1: Clasificación Jerárquica Plana.

[Vásquez, 2022]

Sin embargo, durante la fase de clasificación para cada nuevo ejemplo de prueba el clasificador jerárquico local predice la clase del primer nivel (la clase raíz), después utiliza la clase predicha para limitar las clases a predecir en el siguiente nivel y así sucesivamente hasta llegar a alguno de los nodos hojas de la jerarquía aunque el clasificador también puede finalizar en alguno de los nodos internos de la jerarquía si se definen las reglas necesarias para ello.

El enfoque de clasificación jerárquica local encontramos tres métodos primordiales los cuales son la Clasificación local por nodo, Clasificación local por nodo padre y Clasificación local por nivel [Freitas and de Carvalho, 2007].

3.3.1. Clasificación Jerárquica Local por Nodo

El método de **Clasificación Jerárquica Local por Nodo** basa su funcionamiento en entrenar un clasificador binario por cada nodo de la jerarquía sin tomar en cuenta al nodo raíz. Para el proceso de entrenamiento se debe definir la política con la que se elegirán los objetos válidos y no válidos para el entrenamiento de los clasificadores binarios.

De acuerdo con [Eisner et al., 2005] existen las siguientes políticas:

- **Exclusive.** Se toman como válidos únicamente los objetos pertenecientes al nodo del clasificador, siendo no válidos los objetos restantes.
- **Less exclusive.** Se toman como válidos los objetos pertenecientes al nodo del clasificador exceptuando a los objetos pertenecientes a los nodos descendientes, siendo no válidos los objetos restantes.
- **Less inclusive.** Se toman como válidos los objetos pertenecientes al nodo del clasificador y nodos descendientes, siendo no válidos los objetos restantes.

3. CLASIFICACIÓN JERÁRQUICA

- **Inclusive.** Se toman como válidos los objetos pertenecientes al nodo del clasificador y los de sus nodos descendientes exceptuando a los objetos pertenecientes a la rama del nodo del clasificador, siendo no válidos los objetos restantes.
- **Siblings.** Se toman como válidos los objetos pertenecientes al nodo del clasificador y los de sus nodos descendientes, siendo no válidos los objetos pertenecientes a los nodos hermanos junto con sus respectivos descendientes.

Durante la fase de clasificación cada clasificador decide si el objeto a clasificar pertenece o no a la clase predicha por el clasificador binario. Este método tiene la ventaja de poder predecir múltiples clases por nivel usando para ello algoritmos de clasificación que sólo predicen una etiqueta a la vez, evitando las complicaciones asociadas con el diseño de algoritmos de clasificación para multi-etiquetado.

No obstante, una de sus desventajas es que es propenso al problema de inconsistencia en donde puede darse el caso en que se clasifique erróneamente en algún nodo de los niveles superiores ocasionando que no se llegue siquiera a considerar la clase real a la que pertenece el objeto, acarreando el error de clasificación a los niveles siguientes. Otra desventaja es el problema de bloqueo, el cual sucede siempre y cuando el problema a resolver no requiere que la predicción final sea estrictamente un nodo hoja [Hernández, 2012].

Se representa el enfoque de Clasificación Jerárquica Local por Nodo en la figura (3.2). Los círculos representan las clases de la jerarquía mientras que los recuadros punteados representan los clasificadores binarios. La predicción del clasificador consiste en la ruta indicada por los clasificadores binarios en cada nivel, círculos rojos.

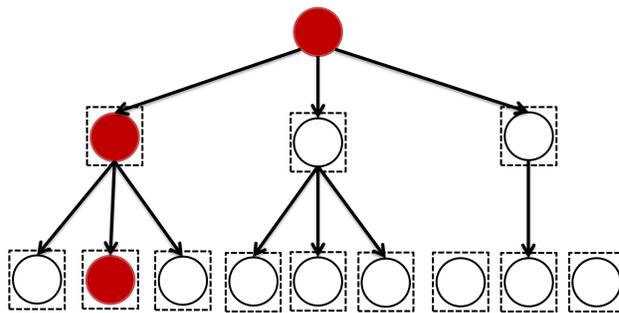


Figura 3.2: Clasificación Jerárquica Local por Nodo.

[Vásquez, 2022]

3.3.2. Clasificación Jerárquica Local por Nodo Padre

El método de **Clasificación Jerárquica Local por Nodo Padre** tiene como objetivo el entrenar un clasificador multi-clase por cada nodo padre presente en la taxonomía, con la finalidad de que se asocie una clase al objeto de acuerdo a los hijos del nodo padre. Se debe elegir con cuidado el algoritmo de clasificación que se utilizara ya que se deberá repetir en cada nodo padre de la jerarquía.

Durante la fase de entrenamiento se entrena cada clasificador de la jerarquía de acuerdo a la política elegida. En la fase de clasificación entonces se comenzará desde el nodo raíz siendo la clase predicha por este nodo la que indique el camino a seguir al siguiente nivel y así sucesivamente hasta llegar a un nodo hoja o, en caso de que lo necesitemos, terminar la predicción en cualquier nodo de la jerarquía.

Se representa el enfoque de Clasificación Jerárquica Local por Nodo Padre en la figura (3.3). Los círculos representan las clases de la jerarquía mientras que los recuadros punteados representan los clasificadores multi-clase. La predicción del clasificador consiste en la ruta indicada por los clasificadores multi-clase en cada nivel, círculos rojos.

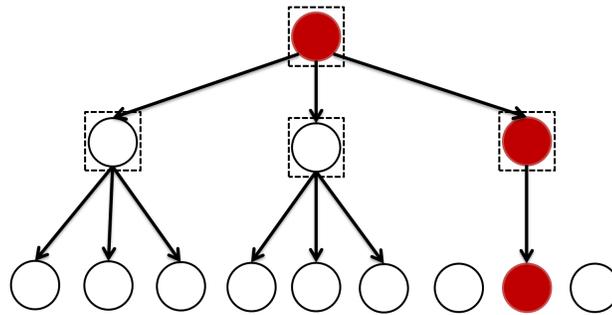


Figura 3.3: Clasificación Jerárquica Local por Nodo Padre.

[Vásquez, 2022]

3.3.3. Clasificación Jerárquica Local por Nivel

El enfoque de **Clasificación Jerárquica Local por Nivel** entrena un clasificador multi-clase por cada nivel en la taxonomía. En la fase de entrenamiento cada clasificador es entrenado con la información local de los nodos de cada nivel de acuerdo a la política de entrenamiento elegida. Por último, en la fase de clasificación se combinan las salidas de los clasificadores en cada nivel.

Las desventajas de este método es que ignora la relación padre-hijo de la jerarquía y es más propenso al problema de inconsistencia. Por otro lado, la ventaja principal de este método es su simplicidad debido al poco uso de clasificadores [Hernández, 2012].

Se representa el enfoque de Clasificación Jerárquica Local por Nivel en la figura (3.4). Los círculos representan las clases de la jerarquía mientras que los recuadros punteados representan los clasificadores multi-clase. La predicción del clasificador consiste en la ruta indicada por los clasificadores multi-clase en cada nivel, círculos rojos.

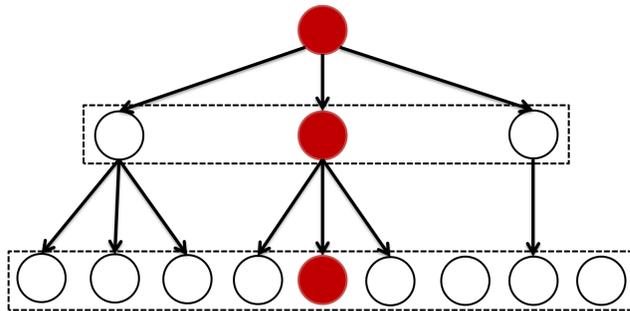


Figura 3.4: Clasificación Jerárquica Local por Nivel.

[Vásquez, 2022]

Comparación Modelos Tradicionales y Jerárquicos

En las siguientes secciones se presentan las investigaciones previas en donde se compararon ambos modelos, algunos proyectos representativos de los mismos y las medidas de evaluación utilizadas para obtener el grado de eficacia de cada modelo. Primeramente debemos recordar las características más importantes de los Modelos Tradicionales y Jerárquicos las cuales se encuentran resumidas en la tabla (4.1).

Modelo Tradicional	<ul style="list-style-type: none">• Un objeto pertenece a una clase única.• No necesita de una taxonomía.• Bajo costo computacional y complejidad.• Se usa un solo clasificador.• Sensible a datos escalados.
Modelo Jerárquico	<ul style="list-style-type: none">• Un objeto pertenece a 1 o más clases.• Hace uso de una taxonomía.• Alto costo computacional y complejidad.• El número de clasificadores depende del método de clasificación y de la taxonomía implementada.

Tabla 4.1: Características de los Modelos Tradicionales y Jerárquicos.

[Hernández, 2012]

4.1. Antecedentes Históricos

Hoy en día, la tarea de buscar trabajos en donde se comparen de forma directa los Modelos Tradicionales y Jerárquicos se encuentra limitados a contados artículos. Pese a la desfavorable situación podemos resaltar una de las investigaciones más recientes perteneciente a [Silva, 2021].

En esta investigación se llegó a la conclusión de que la clasificación jerárquica presenta un mejor rendimiento en dominios jerárquicos (ya sean explícitos o inferidos) en comparación con la clasificación tradicional. Asimismo se propone resolver los problemas de clasificación multi-clase mediante una descomposición jerárquica de acuerdo a una jerarquía de clases inferida por un clasificador tradicional además de presentar métodos en donde se adaptan los modelos a cambios en las clases objetivos.

Por otra parte, en la investigación realizada por [Suca et al., 2016] se compararon los algoritmos de Máquinas de Vectores de Soporte (SVM), las Redes Neuronales Artificiales (RNA), los Árboles de Decisión y Naive Bayes, esto con la finalidad de clasificar la predicción de casos de obesidad infantil. Concluyendo en que los algoritmos de SVM, RNA y los Árboles de Decisión presentan una precisión óptima si la fuente de datos se encuentra pre-procesada, por lo que se descubrieron los clasificadores más adecuados para la predicción de caso de obesidad a partir de los datos de registro médicos de una persona.

Podemos encontrar a través de la web ejercicios de clasificación con fines didácticos en donde se hace uso de Modelos Tradicionales como por ejemplo la predicción de números escritos a mano mediante clasificación de imágenes y la clasificación de 3 tipos de flores basándonos en la longitud de pétalos y sépalos de cada iris [Pedregosa et al., 2011].

Sin embargo, en un ámbito mucho más serio y complejo, la implementación de Modelos Jerárquicos es una alternativa mucho más recurrente. A continuación se presentarán algunas investigaciones de las que a nuestro parecer son las más interesantes.

En el trabajo [Hernández, 2012] se desarrolló un nuevo método de clasificación jerárquico multidimensional en donde se toman en cuenta todas las ramas de la jerarquía para realizar una predicción sin el inconveniente de que se presenten problemas de inconsistencia, problemas recurrentes en los Modelos Jerárquicos.

Además, en el trabajo de [Carmona, 1989] se clasifican los olmos ibéricos haciendo uso de un Modelo Jerárquico tomando como base las variables foliares de la planta. En el trabajo de [López and Fachelli, 2016] se implementó el uso de un Modelo Jerárquico para la clasificación de ajos utilizando para ello las características de los cultivares llegando a clasificarse en 5 clases.

4.2. Medidas de Evaluación

Utilizamos las medidas de evaluación para medir el óptimo rendimiento de nuestro algoritmo de clasificación respecto a los casos que se le presenten y de lo que necesitemos medir. A continuación se mencionan algunas de las más comunes y generalmente utilizadas recopiladas de [Provost and Kohavi, 1998].

Una medida muy utilizada es la exactitud (**accuracy**). La exactitud (4.1) mide la predicción hecha por el modelo sobre la cantidad total de casos.

$$\text{Exactitud} = 1 - (\text{número de errores} / \text{número total de casos}) \quad (4.1)$$

No obstante, esta medida no refleja el desempeño real de nuestro clasificador ya que podríamos estar obteniendo una exactitud bastante buena y a pesar de ello alguna de las clases en nuestro modelo puede ser clasificada incorrectamente. Para dar solución a este inconveniente se creó la **matriz de confusión** la cual es una herramienta que muestra la relación entre las clases predichas y las reales.

Una matriz de confusión tiene un tamaño $L \times L$, donde L es la cantidad de clases. Se presenta un ejemplo en la tabla (4.2) de una matriz de confusión de $L = 2$ donde VP son los verdaderos positivos (veces que el clasificador predijo de forma correcta que un objeto pertenece a esa clase), FP son los falsos positivos (veces que el clasificador predijo que un objeto pertenece a una clase pero estaba equivocado), VN son los verdaderos negativos (veces que el clasificador predijo de forma correcta que un objeto no pertenece a esa clase) y FN son los falsos negativos (veces que el clasificador predijo que un objeto no pertenece a una clase pero estaba equivocado).

Clase Real / Clase predicha	Negativo	Positivo
Negativo	VN	FP
Positivo	FN	VP

Tabla 4.2: Ejemplo de una Matriz de Confusión.

[Vásquez, 2022]

Mediante la matriz de confusión podemos obtener más medidas de evaluación como lo son la **Precisión** y la **Sensibilidad o Recuerdo**.

La precisión (4.2) nos dice cuántos objetos fueron clasificados como positivos y en verdad lo son.

$$\textit{Precisión} = \frac{VP}{VP + FP} \quad (4.2)$$

En cambio, el **Recuerdo o sensibilidad** (4.3) mide lo bien que puede el clasificador reconocer verdaderos positivos.

$$\textit{Recuerdo} = \frac{VP}{VP + FN} \quad (4.3)$$

Mejorar tanto la precisión (4.2) como el recuerdo (4.3) resulta en la disminución de la otra. Es por ello que existe una forma de combinar ambas medidas en una sola, surgiendo así la medida F o **F-Measure** (4.4).

$$\textit{Medida F} = \frac{2 * \textit{Precisión} * \textit{Recuerdo}}{\textit{Precisión} + \textit{Recuerdo}} \quad (4.4)$$

4.3. Análisis de Varianza (ANOVA)

El análisis de la varianza (Analysis Of Variance), permite analizar la variación en una variable de respuesta medida en circunstancias definidas por variables de clasificación o factores [Dagnino, 2014].

La forma más simple es el llamado ANOVA de una vía o factor, cuando existe una sola variable independiente para clasificar a los sujetos y dos o más niveles (que definen los grupos) de ella. Por ejemplo, una variable de clasificación puede ser en qué mes del año hay más flores en un parque.

A grandes rasgos, se calcula la variabilidad total de un set de datos **Suma de Cuadrados** y la de los componentes de las diversas **Fuentes de Variación**. Para ello, se calcula la varianza de las medias de los grupos entre las muestras así como la varianza dentro de las muestras también llamada error o residual.

Cada una de estas variabilidades es reducida a una varianza estimada, conocida como **Cuadrado Medio** o Media cuadrática, dividiéndola por los respectivos **Grados de libertad**, los cuales representan la manera en que el número de muestras y el tamaño de estas entran en las fórmulas matemáticas usadas para calcular un determinado estadístico.

El resultado de ANOVA es la **estadística F** (4.5), en el caso de que F sea mayor al valor critico (valor teórico) se rechazara la hipótesis nula H_0 , lo que significaría que las medias de la población no son iguales. Para el caso contrario, si se rechazara la hipótesis alternativa H_1 , significaría que todas las medias de la población son iguales. La probabilidad de que la hipótesis nula sea verdadera se define como P .

$$F = \frac{CM_{Trat}}{CM_{Error}} \quad (4.5)$$

En la tabla (4.3) el valor critico o F teórico se representa como $F_{\alpha, k-1, N-k}$. Por otra parte la hipótesis nula se definiría de esta forma $H_0 : \mu_1 = \mu_2 = \dots = \mu_k$ siendo k el número de muestras y N el número total de elementos que comprenden todas las muestras medidas. Además, $x_{..}$ corresponde a la suma total de todos los factores independientemente del grupo, n_i es la cantidad de elementos por cada muestra y x_i es la sumatoria de los factores para cada grupo.

Fuentes de Variación	Suma de Cuadrados	Grados de libertad	Media cuadrática
Entre las muestras	$SC_{Trat} = \sum_{i=1}^k (\frac{1}{n_i} * x_i^2) - \frac{x_{..}^2}{N}$	$k - 1$	$CM_{Trat} = \frac{SC_{Trat}}{k-1}$
Dentro de las muestras	$SC_{Error} = SC_T - SC_{Trat}$	$N - k$	$CM_{Error} = \frac{SC_{Error}}{N-k}$
Total	$SC_T = \sum_{i=1}^k \sum_{j=1}^{n_i} x_{ij}^2 - \frac{x_{..}^2}{N}$	$N - 1$	

Tabla 4.3: Tabla para el Análisis de Varianza.

[Vásquez, 2022]

Aunque los valores medios de varios grupos parecen ser diferentes, esto podría ser debido a un error de muestreo más que al efecto de la variable independiente sobre la variable dependiente. Si se debe a un error de muestreo, la diferencia entre las medias de los grupos no tendrá sentido. ANOVA ayuda a averiguar si la diferencia en los valores medios es estadísticamente significativa o si indirectamente una variable independiente está influyendo en alguna variable dependiente muy útil en el aprendizaje automático.

4.4. Método de Turkey

El **Método de Tukey** es una prueba estadística utilizada generalmente en conjunto con ANOVA, consiste en comparar las diferencias entre medias muestrales con el valor crítico dado en la ecuación. Es utilizada en experimentos que implican un número elevado de comparaciones [Canteri et al., 2001].

En la ecuación (4.6) se define al igual que las variables ya establecidas anteriormente, $q_\alpha(k, N - k)$ son los puntos porcentuales de la distribución del rango ya establecido en la tabla (4.1), obteniéndolo ubicando el valor de k en las columnas de la tabla y $N - k$ en las filas.

$$T_\alpha = q_\alpha(k, N - k) \sqrt{\frac{CM_{Error}}{n_i}} \tag{4.6}$$

Tomaremos los pares de medias como significativas si el valor absoluto de la diferencia muestral es mayor que T_α .

$\alpha = 0.05$	n														
m	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
2	6.08	8.33	9.80	10.88	11.73	12.43	13.03	13.54	13.99	14.40	14.76	15.09	15.39	15.67	
3	4.50	5.91	6.82	7.50	8.04	8.48	8.85	9.18	9.46	9.72	9.95	10.15	10.35	10.52	
4	3.93	5.04	5.76	6.29	6.71	7.05	7.35	7.60	7.83	8.03	8.21	8.37	8.52	8.66	
5	3.64	4.60	5.22	5.67	6.03	6.33	6.58	6.80	6.99	7.17	7.32	7.47	7.60	7.72	
6	3.46	4.34	4.90	5.30	5.63	5.90	6.12	6.32	6.49	6.65	6.79	6.92	7.03	7.14	
7	3.34	4.16	4.68	5.06	5.36	5.61	5.82	6.00	6.16	6.30	6.43	6.55	6.66	6.76	
8	3.26	4.04	4.53	4.89	5.17	5.40	5.60	5.77	5.92	6.05	6.18	6.29	6.39	6.48	
9	3.20	3.95	4.41	4.76	5.02	5.24	5.43	5.59	5.74	5.87	5.98	6.09	6.19	6.28	
10	3.15	3.88	4.33	4.65	4.91	5.12	5.30	5.46	5.60	5.72	5.83	5.93	6.03	6.11	
11	3.11	3.82	4.26	4.57	4.82	5.03	5.20	5.35	5.49	5.61	5.71	5.81	5.90	5.98	
12	3.08	3.77	4.20	4.51	4.75	4.95	5.12	5.27	5.39	5.51	5.61	5.71	5.80	5.88	
13	3.06	3.73	4.15	4.45	4.69	4.88	5.05	5.19	5.32	5.43	5.53	5.63	5.71	5.79	
14	3.03	3.70	4.11	4.41	4.64	4.83	4.99	5.13	5.25	5.36	5.46	5.55	5.64	5.71	
15	3.01	3.67	4.08	4.37	4.59	4.78	4.94	5.08	5.20	5.31	5.40	5.49	5.57	5.65	
16	3.00	3.65	4.05	4.33	4.56	4.74	4.90	5.03	5.15	5.26	5.35	5.44	5.52	5.59	
17	2.98	3.63	4.02	4.30	4.52	4.70	4.86	4.99	5.11	5.21	5.31	5.39	5.47	5.54	
18	2.97	3.61	4.00	4.28	4.49	4.67	4.82	4.96	5.07	5.17	5.27	5.35	5.43	5.50	
19	2.96	3.59	3.98	4.25	4.47	4.65	4.79	4.92	5.04	5.14	5.23	5.31	5.39	5.46	
20	2.95	3.58	3.96	4.23	4.45	4.62	4.77	4.90	5.01	5.11	5.20	5.28	5.36	5.43	
21	2.94	3.56	3.94	4.21	4.42	4.60	4.74	4.87	4.98	5.08	5.17	5.25	5.33	5.40	
22	2.93	3.55	3.93	4.20	4.41	4.58	4.72	4.85	4.96	5.06	5.14	5.23	5.30	5.37	
23	2.93	3.54	3.91	4.18	4.39	4.56	4.70	4.83	4.94	5.03	5.12	5.20	5.27	5.34	
24	2.92	3.53	3.90	4.17	4.37	4.54	4.68	4.81	4.92	5.01	5.10	5.18	5.25	5.32	
25	2.91	3.52	3.89	4.15	4.36	4.53	4.67	4.79	4.90	4.99	5.08	5.16	5.23	5.30	
26	2.91	3.51	3.88	4.14	4.35	4.51	4.65	4.77	4.88	4.98	5.06	5.14	5.21	5.28	
27	2.90	3.51	3.87	4.13	4.33	4.50	4.64	4.76	4.86	4.96	5.04	5.12	5.19	5.26	
28	2.90	3.50	3.86	4.12	4.32	4.49	4.62	4.74	4.85	4.94	5.03	5.11	5.18	5.24	
29	2.89	3.49	3.85	4.11	4.31	4.47	4.61	4.73	4.84	4.93	5.01	5.09	5.16	5.23	
30	2.89	3.49	3.85	4.10	4.30	4.46	4.60	4.72	4.82	4.92	5.00	5.08	5.15	5.21	
31	2.88	3.48	3.84	4.09	4.29	4.45	4.59	4.71	4.81	4.90	4.99	5.06	5.13	5.20	
32	2.88	3.48	3.83	4.09	4.28	4.45	4.58	4.70	4.80	4.89	4.98	5.05	5.12	5.18	
33	2.88	3.47	3.83	4.08	4.28	4.44	4.57	4.69	4.79	4.88	4.97	5.04	5.11	5.17	
34	2.87	3.47	3.82	4.07	4.27	4.43	4.56	4.68	4.78	4.87	4.96	5.03	5.10	5.16	
35	2.87	3.46	3.81	4.07	4.26	4.42	4.56	4.67	4.77	4.86	4.95	5.02	5.09	5.15	
36	2.87	3.46	3.81	4.06	4.25	4.41	4.55	4.66	4.76	4.85	4.94	5.01	5.08	5.14	
37	2.87	3.45	3.80	4.05	4.25	4.41	4.54	4.66	4.76	4.85	4.93	5.00	5.07	5.13	
38	2.86	3.45	3.80	4.05	4.24	4.40	4.53	4.65	4.75	4.84	4.92	4.99	5.06	5.12	
39	2.86	3.45	3.79	4.04	4.24	4.39	4.53	4.64	4.74	4.83	4.91	4.98	5.05	5.11	
40	2.86	3.44	3.79	4.04	4.23	4.39	4.52	4.63	4.73	4.82	4.90	4.98	5.04	5.11	
41	2.86	3.44	3.79	4.03	4.23	4.38	4.51	4.63	4.73	4.82	4.90	4.97	5.04	5.10	
42	2.85	3.44	3.78	4.03	4.22	4.38	4.51	4.62	4.72	4.81	4.89	4.96	5.03	5.09	
43	2.85	3.43	3.78	4.03	4.22	4.37	4.50	4.62	4.72	4.80	4.88	4.96	5.02	5.08	
44	2.85	3.43	3.78	4.02	4.21	4.37	4.50	4.61	4.71	4.80	4.88	4.95	5.02	5.08	
45	2.85	3.43	3.77	4.02	4.21	4.36	4.49	4.61	4.70	4.79	4.87	4.94	5.01	5.07	
46	2.85	3.42	3.77	4.01	4.20	4.36	4.49	4.60	4.70	4.79	4.87	4.94	5.00	5.06	
47	2.85	3.42	3.77	4.01	4.20	4.36	4.48	4.60	4.69	4.78	4.86	4.93	5.00	5.06	
48	2.84	3.42	3.76	4.01	4.20	4.35	4.48	4.59	4.69	4.78	4.86	4.93	4.99	5.05	
49	2.84	3.42	3.76	4.00	4.19	4.35	4.48	4.59	4.69	4.77	4.85	4.92	4.99	5.05	
50	2.84	3.42	3.76	4.00	4.19	4.34	4.47	4.58	4.68	4.77	4.85	4.92	4.98	5.04	

Figura 4.1: Tabla de Cuantiles de la Distribución de Tukey.

[Mota, 2010]

Construcción de los Modelos Tradicionales y Jerárquico

En este capítulo se describen de forma detallada los modelos de clasificación implementados en la presente tesis explicando de forma general el código fuente de los programas y presentando diagramas de flujo para facilitar su entendimiento. La construcción y ajuste de los modelos se realizó mediante el uso de la API *Scikit-learn* [Pedregosa et al., 2011] teniendo como base al lenguaje de programación *Python*.

Asimismo, se pueden acceder al código fuente de los programas, metadatos y datasets en este repositorio de **Github**:

<https://github.com/Dxvid1000/ModelosTradicionalesJerarquicos.git>.

A pesar de los esfuerzos para disminuir el tiempo que toma el entrenamiento y prueba en cada ejecución del código de los modelos, este mismo sigue siendo alto para el caso del modelo jerárquico. Es por ello que se necesitó de una computadora con una gran potencia de cómputo para la finalidad de agilizar este proceso.

Las especificaciones de la computadora en donde se realizaron las pruebas y construcción de dichos modelos son las siguientes: Procesador Ryzen 5 3600 6-Core, 16 GB de RAM y Sistema Operativo Windows 10 Education.

5.1. Preparación del DataSet

Buscar a través de la red datasets especializados en el idioma español no es una tarea nada sencilla debido a su poco o casi nulo interés en el estudio de esta lengua, el requerimiento de una suscripción de paga bastante generosa o no son realmente representativos.

Afortunadamente se encontró un corpus de datos que representa adecuadamente el idioma español de México y que a su vez es de libre acceso para propósitos académicos, nos referimos al **Corpus DIMEx100** <https://turing.iimas.unam.mx/~luis/DIME/CORPUS-DIMEX.html>, creado y distribuido por la Universidad Nacional Autónoma de México en el departamento de Ciencias de la Computación del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS).

El DIMEx100 hace uso de tres niveles de transcripción fonética los cuales son el nivel T54, T44 y el T22, este último utilizado en la presente tesis. Véase el apéndice B.

Empleando el corpus de datos DIMEx100 se procedió a extraer las representaciones alofónicas pertenecientes al nivel T22 en archivos de audio de formato *WAV* para posteriormente agruparlas en clases de acuerdo al alófono que representa. Resultando así en un total de 268,148 datos totales y 22 clases (una por cada alófono). Se designó un 20 % del total de nuestros datos para el dataset de entrenamiento y el 80 % restante le pertenece al dataset de prueba. Véase el apéndice C.

5.2. Preprocesamiento de los datos

Para una óptima y sencilla representación de los features de nuestros objetos se optó por representar el feature como el promedio de calcular el **Mel Frequency Cepstrum Coefficient** (MFCC) 40 veces con una longitud de ventana FFT (versión optimizada para el cálculo computacional de la transformada discreta de Fourier) de 512, utilizando para ello la función *librosa.feature.mfcc()* de la librería *librosa* [McFee et al., 2021].

Debido al tiempo elevado que lleva el cálculo del MFCC para cada objeto de nuestros dataset de prueba y entrenamiento se recurrió al uso de metadatas y programación paralela utilizando para ello el paquete *multiprocessing* desarrollado por Python Software Foundation, dividiendo así la tarea de conseguir los features MFCC pertenecientes a cada objeto en 8 procesos que trabajaran de forma simultánea utilizando los objetos previamente divididos en archivos llamados metadatas.

Por último, escalamos nuestros features gracias a la función *StandardScaler()* incluida en Scikit-learn, esto con el fin de mejorar el desempeño y velocidad de nuestros modelos. En la figura (5.1) podemos tener una idea general de cuál es el funcionamiento y los pasos necesarios en la ejecución de los programas correspondientes a nuestros modelos.

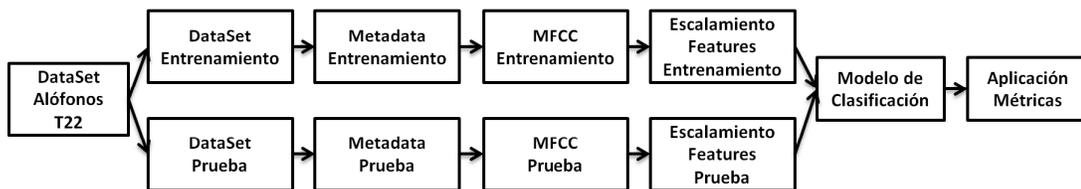


Figura 5.1: Funcionamiento General de los Modelos de Clasificación.

[Vásquez, 2022]

5.3. Creación y Ajuste de los Modelos Tradicionales

El paso siguiente es construir y ajustar nuestros modelos para conseguir su máximo rendimiento. En el caso del MLP utilizamos la función *MLPClassifier()* incluida en Scikit-learn. Los parámetros que muestran la mayor precisión son el solver “adam“, un “alpha“ de $2e-5$, una capa de entrada de 40, junto con una capa intermedia de 120 y un máximo de iteraciones de 300. El parámetro “random_state“ es usado para mantener el mismo modelo para la fase de prueba a través de múltiples ejecuciones.

```
MLPClassifier(solver = 'adam', alpha = 2e-5, hidden_layer_sizes = (40,120)
, max_iter=300, random_state = 42)
```

Para el caso de la SVM utilizamos la función *svm.SVC()* incluida en Scikit-learn. Respectivamente, los parámetros que muestran la mayor precisión son el kernel “rbf“, un gamma “auto“, un “C“ de 5.0 y un máximo de iteraciones de 10,000. El parámetro random_state es usado para el mismo fin que en el MLP.

```
svm.SVC(kernel = 'rbf', max_iter = 10000, gamma = 'auto', C = 5.0,
random_state = 42)
```

5. CONSTRUCCIÓN DE LOS MODELOS TRADICIONALES Y JERÁRQUICO

Finalmente, se entrena nuestro modelo utilizando para ello los features escalados junto con los targets de entrenamiento. Terminando en la predicción de los features de prueba igualmente escalados.

```
.fit(X_trainscaled, y_train).predict(X_testscaled)
```

Debido a la inmensa cantidad de combinaciones posibles entre estos parámetros, no se descarta que se puedan obtener un rendimiento mejor, aunque se ha notado a lo largo de las iteraciones previas que dicho aumento no es realmente significativo.

5.4. Creación y Ajuste del Modelo Jerárquico

El modelo jerárquico construido en la presente tesis toma el enfoque de **clasificación local por nodo padre**. Se optó por una política **Siblings** para la fase de entrenamiento de nuestros nodos Padres y Raíz y una profundidad de nodos hojas.

A continuación se presenta la taxonomía perteneciente a nuestro modelo jerárquico, donde los círculos representan a los nodos y los recuadros punteados representan a los clasificadores binarios, siendo 24 en total:

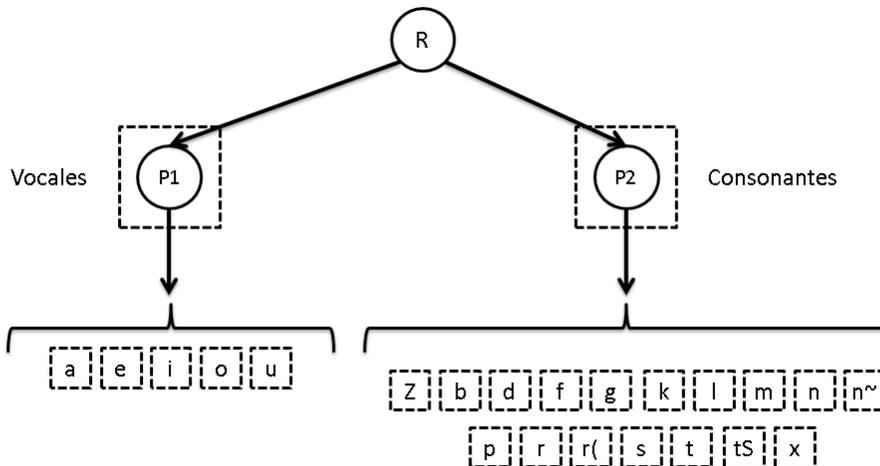


Figura 5.2: Taxonomía del Modelo Jerárquico.

[Vásquez, 2022]

Como se puede observar en la figura (5.2) se dividieron los datos en subclases ya que se observó que algunos alófonos son sumamente parecidos a otros o a veces son pronunciados de forma distinta, lo que ocasionaba que nuestro modelo clasificara erróneamente. Dentro del **nodo Raíz** se encuentran las clases “Vocales” y “Consonantes”. En el **nodo Padre 1** tenemos las clases “a”, “e”, “i”, “o” y “u”. Por último, dentro en el **nodo Padre 2** tenemos las clases “Z”, “b”, “d”, “f”, “g”, “k”, “l”, “m”, “n”, “n ”, “p”, “r”, “r(”, “s”, “t”, “tS” y “x”.

Los clasificadores binarios utilizados en el modelo jerárquico serán perceptrones multicapa y contarán con los parámetros de un solver “adam”, un “alpha” de $2e - 5$, una única capa de 40, un máximo de iteraciones de 300 y un parámetro “random_state” utilizado para mantener el mismo modelo para la fase de prueba a través de múltiples ejecuciones.

```
MLPClassifier(solver = 'adam', alpha = 2e-5, hidden_layer_sizes = (40),
              max_iter=300, random_state = 42)
```

Para que el modelo inicie el proceso de clasificación se llama a la función *predecirAlofonos()* en donde es necesario ingresar la ruta de las metadatos de prueba ya que esta misma comienza con la creación y entrenamiento de cada uno de los clasificadores binarios retornando con los targets predichos y de prueba.

```
predecirAlofonos('Metadata22/Test')
```

Se buscó de forma intencional una sobrecarga de datos en una determinada clase en cada uno de los clasificadores binarios que conforman a cada nodo con el fin de que cada uno de los clasificadores binarios se encargue únicamente de clasificar si pertenecen o no a la clase con la que fueron sobrecargados. Debido a esto es que se crearon nuevos datasets de entrenamiento para cada clasificador binario. Por ejemplo, en el nodo Raíz, en donde podemos encontrar dos clasificadores binarios (vocales y consonantes). Se entrenó un clasificador binario con una carga hacia la clase “Vocales” y el otro clasificador con una carga a la clase “Consonantes”.

Obtención de Resultados

En este capítulo se presentaran las precisiones y el tiempo obtenidos después de 30 ejecuciones para un posterior análisis de varianza en cada modelo, esto permitirá obtener el grado de eficacia de cada modelo.

6.1. Experimentación

Para medir la precisión de nuestros modelos hicimos uso de la **medida F**, la cual se implementa mediante la función `accuracy_score()` y que requiere de los parámetros “y_test” (clase verdadera) y “y_pred” (clase predicha). Cabe aclarar que se mantuvieron sin cambios los datos de entrenamiento y prueba así como los parámetros en las 30 ejecuciones de nuestros modelos.

```
print('PRECISION_MODELO:', accuracy_score(y_test, y_pred) * 100)
```

6.2. Precisión en el Reconocimiento de Alófonos de los Modelos Tradicionales

Se presentan las precisiones y el tiempo obtenidos después de 30 ejecuciones de los modelos tradicionales “Perceptrón Multicapa” (MLP) representadas mediante la tabla [6.1](#).

6. OBTENCIÓN DE RESULTADOS

#	Precisión	Tiempo	#	Precisión	Tiempo	#	Precisión	Tiempo
1	49.14	10.02	11	49.45	6.84	21	50.50	9.48
2	49.74	6.57	12	50.14	7.14	22	50.22	6.40
3	49.47	6.77	13	50.52	6.56	23	49.61	6.61
4	50.23	6.39	14	51.04	6.55	24	49.58	6.35
5	50.23	6.39	15	54.48	8.94	25	51.28	6.64
6	49.76	6.38	16	50.67	9.47	26	51.04	6.47
7	49.58	6.89	17	49.69	6.51	27	50.08	6.36
8	50.31	6.78	18	50.52	6.37	28	49.85	6.32
9	49.61	6.53	19	49.45	6.55	29	51.28	6.47
10	50.40	6.49	20	50.05	6.45	30	50.52	6.64

Tabla 6.1: Precisión en el Reconocimiento de Alófonos de un MLP.

[Vásquez, 2022]

Asimismo se presentan las precisiones y el tiempo obtenidos de las “Máquinas de Vectores de Soporte” (SVM) presentadas mediante la tabla 6.2.

#	Precisión	Tiempo	#	Precisión	Tiempo	#	Precisión	Tiempo
1	52.95	49.16	11	52.95	44.91	21	52.95	42.71
2	52.95	44.73	12	52.95	44.12	22	52.95	39.96
3	52.95	50.68	13	52.95	46.82	23	52.95	43.08
4	52.95	44.71	14	52.95	45.84	24	52.95	40.05
5	52.95	48.34	15	52.95	42.97	25	52.95	43.00
6	52.95	52.05	16	52.95	40.09	26	52.95	40.06
7	52.95	50.86	17	52.95	42.50	27	52.95	42.66
8	52.95	45.47	18	52.95	40.03	28	52.95	42.80
9	52.95	43.06	19	52.95	40.35	29	52.95	42.62
10	52.95	42.70	20	52.95	40.32	30	52.95	40.06

Tabla 6.2: Precisión en el Reconocimiento de Alófonos de un SVM.

[Vásquez, 2022]

6.3. Precisión en el Reconocimiento de Alófonos de un Modelo Jerárquico

Se presentan las precisiones basadas en la medida F y el tiempo en minutos obtenidos durante las 30 ejecuciones del modelo jerárquico representadas mediante la tabla 6.3. Se puede observar el número correspondiente a la ejecución junto con la precisión obtenida y el tiempo que tomó llegar a la misma.

#	Precisión	Tiempo	#	Precisión	Tiempo	#	Precisión	Tiempo
1	100.00	68.80	11	100.00	68.35	21	100.00	66.49
2	100.00	64.84	12	100.00	67.20	22	100.00	70.31
3	100.00	64.93	13	100.00	63.08	23	100.00	64.84
4	100.00	62.83	14	100.00	64.61	24	100.00	62.87
5	100.00	67.86	15	100.00	62.44	25	100.00	62.99
6	100.00	64.46	16	100.00	61.81	26	100.00	67.10
7	100.00	67.09	17	100.00	65.11	27	100.00	70.64
8	100.00	65.12	18	100.00	64.51	28	100.00	67.16
9	100.00	62.41	19	100.00	64.40	29	100.00	64.55
10	100.00	62.97	20	100.00	65.26	30	100.00	65.24

Tabla 6.3: Precisión en el Reconocimiento de Alófonos de un Modelo Jerárquico.

[Vásquez, 2022]

6.4. Analisis de Varianza para los Modelos Tradicionales y Jerárquico

Se realizara un análisis de varianza de una vía, en este caso la variable independiente será la precisión de nuestros modelos dejando al lado el tiempo de ejecución debido a que podrían influir factores externos en el mismo. Debido a esto se contara con tres grupos MLP, SVM y JERARQUICO.

6. OBTENCIÓN DE RESULTADOS

Inicialmente contamos con los siguientes datos:

- $F_{\alpha, k-1, N-k}$: valor critico o F teórico = 3.101295757.
- k : Número de muestras = 30.
- N : Número total de elementos = 90.
- $x_{..}$: Suma total de todos los factores = 6096.94.
- n_i : Cantidad de elementos por muestra MLP = 30, SVM = 30 y JERARQUICO = 30.
- x_i : Sumatoria de factores para cada grupo MLP = 1508.44, SVM = 1588.5 y JERARQUICO = 3000.

Realizando los cálculos correspondientes al análisis de varianza (6.4) se obtiene la siguiente tabla.

Fuentes de Variación	Suma de Cuadrados	Grados de libertad	Media cuadrática
Entre las muestras	$SC_{Trat} = 46927.70097$	2	$CM_{Trat} = 23463.8504$
Dentro de las muestras	$SC_{Error} = 27.61114667$	87	$CM_{Error} = 0.317369502$
Total	$SC_T = 46955.31212$	89	

Tabla 6.4: Tabla para el Análisis de Varianza de los Modelos Tradicionales y Jerárquico.

[Vásquez, 2022]

De acuerdo con esta tabla se tiene un $F = 73932.27876$ (Estadística F) y un $P = 2.94E^{-141}$ (Probabilidad de que la hipótesis nula sea verdadera). Ya que F es mayor al valor critico significa que se rechazara la hipótesis nula H_0 , por lo tanto existen diferencias entre las medias de los tratamientos de los grupos MLP, SVM y JERARQUICO.

6.5. Aplicación del Método de Tukey

Se realizaron los cálculos correspondientes al método de Tukey tomando en cuenta los valores anteriores. Dichos cálculos pueden observarse en la ecuación 6.1.

$$\begin{aligned}
 T_\alpha &= q_\alpha(k, N - k) \sqrt{\frac{CM_{Error}}{n_i}} \\
 T_\alpha &= q_\alpha(30, 90 - 30) \sqrt{\frac{0.317369502}{30}} \\
 T_\alpha &= q_\alpha(30, 60) * 0.010578983 \\
 T_\alpha &= 4.75 * 0.010578983 \\
 T_\alpha &= 0.488557379
 \end{aligned} \tag{6.1}$$

Acorde con la tabla (6.5) los grupos MLP, SVM y JERARQUICO presentan una diferencia significativa.

Diferencia Poblacional	Diferencia Muestral	Decisión
$\mu_{MLP} - \mu_{SVM}$	2.668666667	Diferencia Significativa
$\mu_{MLP} - \mu_{JERÁRQUICO}$	49.71866667	Diferencia Significativa
$\mu_{SVM} - \mu_{JERÁRQUICO}$	47.05	Diferencia Significativa

Tabla 6.5: Tabla para el Análisis de Varianza de los Modelos Tradicionales y Jerárquico.

[Vásquez, 2022]

Conclusión

7.1. Resumen

La finalidad de esta tesis fue comprobar la hipótesis de que los Modelos Jerárquicos presentan una mejor efectividad en comparación con los Modelos Tradicionales en el ámbito del reconocimiento del lenguaje Español hablado en México. Inicialmente se llevó a cabo la comparación de las eficacias obtenidas mediante la *medida F*, estos resultados se tomaron como el promedio calculado de 30 ejecuciones correspondientes a los Modelos Tradicionales y Jerárquicos.

Los Modelos Tradicionales utilizados para este proyecto de tesis fueron las *Máquinas de Soporte Vectoriales* y *Perceptrón Multicapa* esto por su gran rendimiento y sencillez en el proceso de clasificación, creación y ajuste de los mismos. El Modelo Jerárquico evaluado se trata de un Modelo Jerárquico Local por Nodo Padre. Se aplicó una política de entrenamiento *Siblings*, clasificadores binarios (siendo estos perceptrones multicapa) y una taxonomía construida respecto a los 22 alófonos que conforman al idioma Español Mexicano.

Además se buscó de forma intencional una sobrecarga de datos en cada Nodo Padre con el fin de que cada clasificador se encargue de clasificar si pertenecen o no a su clase encargada asignando una clase al objeto de acuerdo a la clase con mayor probabilidad de todas, repitiendo el mismo proceso hasta llegar a un Nodo Hoja.

Por otra parte, se buscó una representación con la cual poder manipular de forma eficiente y práctica nuestro dataset. Es por ello que se optó por crear representaciones *MFCC* de cada objeto. Subsecuentemente, se implementó programación paralela a la creación de los MFCC y de archivos llamados *Metadatas* con el objetivo de disminuir el tiempo de ejecución de cada programa. Finalmente, se escalaron los *features* de los datasets de entrenamiento y prueba antes de ser pasados a los modelos para su respectiva clasificación.

7.2. Conclusiones

Los experimentos realizados en este trabajo de tesis para la clasificación de los 22 alófonos principales que conforman al Español Mexicano se resumen en una eficacia obtenida mediante la métrica F1 de 50.28 % y 52.95 % para los Modelos Tradicionales Máquinas de Soporte Vectoriales y Perceptrón Multicapa correspondientemente. En el caso del Modelo Jerárquico Local por Nodo Padre se obtuvo una precisión del 100 %.

Cabe resaltar que dichas precisiones son el resultado de aplicar el promedio a las eficacias obtenidas después de 30 ejecuciones manteniendo el mismo modelo en cada una con el fin de cumplir con una comprobación estadística manteniendo la fiabilidad de nuestros resultados.

De acuerdo con los resultados obtenidos, se comprueba la hipótesis de que los Modelos Jerárquicos presentan una mayor precisión en comparación con los Modelos Tradicionales. Sin embargo se presentaron algunas características importantes consecuencia de la implementación de dichos modelos presentados a continuación:

1. El tiempo promedio que toma la ejecución del modelo MLP es mucho menor (6.94 minutos) comparado con el requerido por las SVM (43.89 minutos). Sin embargo ambos modelos requieren de mucho menos tiempo que el Modelo Jerárquico (65.34 minutos).
2. La tarea que más consume tiempo durante la ejecución del modelo Jerárquico es la creación de los MFCC para cada objeto de nuestro dataset. Esto se debe a que en cada Nodo Padre se deben de entrenar sus respectivos clasificadores binarios siendo 22 en total.
3. Debido a errores de grabación en la creación del corpus de datos utilizado, errores de pronunciación cometidos por las personas que prestaron su voz o una gran similitud en el sonido de algunos alófonos de nuestro dataset (por ejemplo tenemos los alófonos [x] y [tS]) es que nuestro Modelos Jerárquico presenta un mejor rendimiento en comparación con los Modelos Tradicionales ya que el mismo aísla los posibles errores de clasificación gracias a su taxonomía.

En general, el Modelo Jerárquico presenta una eficacia superior a los Modelos Tradicionales aquí evaluados para el caso de la clasificación de los 22 alófonos correspondientes al nivel más básico del Español hablado en México pero con la desventaja de recurrir a un tiempo muy grande como lo es casi una hora de ejecución incluso siendo ejecutado en una computadora con notables características.

7.3. Aportaciones y Contribuciones

Del mismo modo que se comprobó que un Modelo Jerárquico Local por Nodo Padre presenta una eficacia acorde con la métrica F1 superior a las obtenidas por los Modelos Tradicionales Máquinas de Soporte Vectoriales y Perceptrón Multicapa para el caso de la clasificación de los 22 alófonos principales del español hablado en México a su vez se contribuyó con un modelo eficaz para convertirse en la base de futuros reconocedores del habla combinando el mismo con otros algoritmos de acuerdo al caso de uso.

Dicho modelo puede aplicarse en los casos en que se desee clasificar los alófonos obtenidos en una conversación en tiempo real o archivos de audio, para su posterior procesamiento o transcripción a texto plano.

7.4. Trabajo a Futuro

Como trabajo a futuro presentamos los siguientes puntos:

1. Poner a prueba los Modelos Tradicionales y Jerárquicos en un ambiente real.
2. Implementar los tipos restantes de Modelos Jerárquicos (Local por Nodo y Local por Nivel) para el reconocimiento de los 22 alófonos principales del español.
3. Llevar a cabo la creación de un reconocedor del habla del lenguaje español en tiempo real, en donde se implemente el Modelo Jerárquico creado en la presente tesis.
4. Crear los Modelos Jerárquicos correspondientes a los niveles T44 y T54 correspondientes a los fonemas del Español de México.
5. Comparar las eficacias de distintos Modelos Tradicionales, Naive Bayes, Árboles de Decisión por mencionar solo algunos, en contraste con nuestro Modelo Jerárquico.
6. Encontrar la forma de disminuir el tiempo que lleva la creación de las representaciones MFCC de los objetos de nuestro dataset en ambos modelos.

Bibliografía

- [Abramowitz and Stegun, 1972] Abramowitz, M. and Stegun, I. (1972). Handbook of mathematical functions with formulas, graphs, and mathematical tables. *Dover*, 9:83–86. [6](#)
- [Canteri et al., 2001] Canteri, M., Althaus, R., Virgens, F., Giglioti, E., and Godoy, C. (2001). Sistema para análise e separação de médias em experimentos agrícolas pelos métodos scott. [22](#)
- [Carmona, 1989] Carmona, R. (1989). *Contribución a la taxonomía numérica de los olmos ibéricos a través de algoritmos de clasificación jerárquica*. PhD thesis, Universidad Politécnica de Madrid. [18](#)
- [Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297. [9](#)
- [Dagnino, 2014] Dagnino, J. (2014). Análisis de varianza. *Revista chilena de anestesia*, 43(4):306–310. [20](#)
- [Diaz and Ramirez, 2018] Diaz, C. and Ramirez, M. (2018). El alófono, el fonema i (conceptualización). *Profundización en Fonología*, pages 2–10. [2](#)
- [Eisner et al., 2005] Eisner, R., Poulin, B., Szafron, D., Lu, P., and Greiner, R. (2005). Improving protein function prediction using the hierarchical structure of the gene ontology. *Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 354–363. [13](#)
- [Freitas and de Carvalho, 2007] Freitas, A. and de Carvalho, A. (2007). A tutorial on hierarchical classification with applications in bioinformatics. *Research and Trends in Data Mining Technologies and Applications*, pages 175–208. [12](#), [13](#)
- [Hahnloser et al., 2000] Hahnloser, R., Sarpeshkar, R., Mahowald, M., Douglas, R., and Seung, H. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405:947–951. [6](#)
- [Haykin and Network, 2004] Haykin, S. and Network, N. (2004). A comprehensive foundation. *Neural networks*, 2(2004):41. [5](#), [6](#), [8](#)

BIBLIOGRAFÍA

- [Hernández, 2012] Hernández, J. (2012). Clasificación jerárquica multidimensional. Master’s thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica. [1](#), [5](#), [6](#), [11](#), [12](#), [14](#), [16](#), [17](#), [18](#)
- [Koehrsen, 2018] Koehrsen, W. (2018). Overfitting vs. underfitting: A complete example. *Towards Data Science*. [6](#)
- [Koller and Sahami, 1997] Koller, D. and Sahami, M. (1997). Hierarchically classifying documents using very few words. *International Conference on Machine Learning*, pages 170–178. [12](#)
- [López and Fachelli, 2016] López, R. and Fachelli, S. (2016). Análisis de clasificación. *Metodología de la investigación social cuantitativa*, pages 5–101. [18](#)
- [McFee et al., 2021] McFee, B., Metsai, A., McVicar, M., Balke, S., Thomé, C., Raffel, C., Zalkow, F., Malek, A., Dana, Lee, K., Nieto, O., Ellis, D., Mason, J., Battenberg, E., Seyfarth, S., Yamamoto, R., viktorandreevichmorozov, Choi, K., Moore, J., Bittner, R., Hidaka, S., Wei, Z., nullmightybofo, Hereñú, D., Stöter, F.-R., Friesch, P., Weiss, A., Vollrath, M., Kim, T., and Thassilo (2021). Librosa. [24](#)
- [Mota, 2010] Mota, M. (2010). <http://matematicas.unex.es>. [22](#)
- [Pedregosa et al., 2013] Pedregosa, F., Buitinck, L., Louppe, G., Blondel, M., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). Api design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122. [7](#), [8](#), [9](#)
- [Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830. [7](#), [18](#), [23](#)
- [Pratheeksha, 2018] Pratheeksha, N. (2018). <https://medium.com/>. [3](#)
- [Provost and Kohavi, 1998] Provost, F. and Kohavi, R. (1998). Glossary of terms. *Journal of Machine Learning*, 30(2-3):271–274. [2](#), [3](#), [19](#)
- [Rebollo, 1981] Rebollo, M. (1981). ¿fonema o rasgo pertinente? [2](#)
- [Rojo, 2016] Rojo, G. (2016). Los corpus textuales del español. *J. Gutierrez-Rexach*. [2](#)
- [Silva, 2021] Silva, P. (2021). *Clasificación jerárquica multiclase*. PhD thesis, Universitat Politècnica de València. [18](#)

- [Suca et al., 2016] Suca, C., Córdova, A., Condori, A., Cayra, J., and Sulla, J. (2016). Comparación de algoritmos de clasificación para la predicción de casos de obesidad infantil. *Universidad Nacional de San Agustín*. 18
- [Tiwari, 2010] Tiwari, V. (2010). Mfcc and its applications in speaker recognition. *International journal on emerging technologies*, 1(1):19–22. 3
- [Van, 1986] Van, C. (1986). *Frank Rosenblatt: Principles of Neurodynamics: Perceptions and the Theory of Brain Mechanisms*. Springer. 6
- [Vásquez, 2022] Vásquez, D. (2022). Efectividad de los modelos jerárquicos y tradicionales para el reconocimiento del habla en el lenguaje español de México. *Universidad Nacional Autónoma de México*. 1, 13, 14, 15, 16, 19, 21, 25, 26, 30, 31, 32, 33, 45
- [Wu et al., 2005] Wu, F., Zhang, J., and Honavar, V. (2005). Learning classifiers using hierarchically structured class taxonomies. *Proceedings of the 6th international conference on Abstraction, Reformulation and Approximation*, pages 313–320. 11
- [Álvarez, 2020] Álvarez, F. (2020). <https://academialatin.com/>. 3

Lista de Variables

A continuación se listan las variables utilizadas a lo largo de la presente tesis:

- \mathbf{x} : Objeto a clasificar.
- n : Cantidad de objetos de entrenamiento.
- m : Cantidad de features de un objeto o vector de características.
- $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$: Set de features los cuales representan las características de un objeto.
- l / \mathbf{y} : Clase o target a la que pertenece un objeto.
- L : Conjunto finito de clases.
- $f(\cdot)$: Función de aprendizaje usada en un perceptrón multicapa.
- i : Cantidad de neuronas de entrada en un MLP.
- $\{\mathbf{x}_i | \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$: Set de neuronas representadas en la capa de entrada.
- a : Cantidad de capas ocultas en un MLP.
- h : Cantidad de neuronas en cada capa oculta en un MLP.
- $\varphi(\cdot)$: Función de activación no lineal.
- $w_1x_1 + w_2x_2 \dots w_nx_n$: Pesos correspondientes a cada neurona.
- o : Cantidad de neuronas de salida en un MLP.
- $f(\mathbf{X})$: Escalar de salida.
- C : Parámetro C.

A. LISTA DE VARIABLES

- γ : Parámetro gamma.
- C_T : Conjunto finito de todas las clases en el dominio de aplicación.
- \prec : Relación anti-reflexiva y transitiva.
- VP : Verdaderos Positivos.
- FP : Falsos Positivos.
- VN : Verdaderos Negativos.
- FN : Falsos Negativos.
- F : Estadística F.
- $F_{\alpha, k-1, N-k}$: valor crítico o F teórico.
- H_0 : Hipótesis nula.
- H_1 : Hipótesis alternativa.
- k : Número de muestras.
- N : Número total de elementos.
- $x_{..}$: Suma total de todos los factores.
- n_i : Cantidad de elementos por muestra.
- x_i : Sumatoria de factores para cada grupo.
- P : Probabilidad de que la hipótesis nula sea verdadera.
- SC_{Trat} : Suma de cuadrados del tratamiento.
- SC_{Error} : Suma de cuadrados del error.
- SC_T : Suma de cuadrados del total.
- CM_{Trat} : Media cuadrática del tratamiento.
- CM_{Error} : Media cuadrática del error.
- $q_{\alpha}(k, N - k)$: Puntos porcentuales de la distribución establecidos en la tabla de cuantiles de la distribución de Tukey.

Alofónos Nivel T22

Representación de las 22 principales formas alofónicas del Español de México dentro del Corpus DIMEx100.

Consonantes	Labiales	Labiodentales	Dentales	Alveolares	Palatales	Velares
Oclusivas sordas	p		t			k
Oclusivas sonoras	b		d			g
Africada sorda					tS	
Fricativas sordas		f		s		x
Fricativas sonora					Z	
Nasales	m			n	n	
Vibrantes				r(/ r		
Lateral				l		

Tabla B.1: Consonantes del Nivel T22 en el Corpus DIMEx100.

Vocales	Anteriores	Central	Posteriores
Cerradas	i		u
Medias	e		o
Abierta		a	

Tabla B.2: Vocales del Nivel T22 en el Corpus DIMEx100.

[Vásquez, 2022]

Bitácora del DataSet

A continuación se explica de forma detallada la distribución de los datos correspondientes a los datasets que fueron utilizados en la construcción de los modelos de clasificación de este proyecto de tesis.

Total de datos en el dataset: 268,148.

Total para Test (80 %): 214,519. **Total para Train (20 %):** 53,629.

Distribución de datos para **prueba**. Los mismos datos fueron evaluados tanto en los **modelos tradicionales** como en el jerárquico.

[Z]: 397, [a]: 26555, [b]: 3320, [d]: 11850, [e]: 31359, [f]: 807, [g]: 999, [i]: 18385, [k]: 7865, [l]: 12216, [m]: 5417, [n]: 15784, [n~]: 116, [o]: 21042, [p]: 4630, [r]: 695, [r(): 12946, [s]: 24806, [t]: 10373, [tS]: 210, [u]: 4130, [x]: 617

Distribución de datos para **entrenamiento** de los **modelos tradicionales**:

[Z]: 600, [a]: 4230, [b]: 2500, [d]: 2500, [e]: 4230, [f]: 1500, [g]: 1600, [i]: 4230, [k]: 2500, [l]: 2500, [m]: 2500, [n]: 2500, [n~]: 320, [o]: 4230, [p]: 2500, [r]: 1500, [r(): 2500, [s]: 2500, [t]: 2500, [tS]: 359, [u]: 4230, [x]: 1600

Distribución de datos para **entrenamiento** de acuerdo al nodo en el **modelo jerárquico**:

- **Nodo Raíz**

- **Vocales:** 26,250.
 - Vocales: 4,230 cada una. 21,150 en total.
 - Consonantes: 300 cada uno. 5,100 en total.

- **Consonantes:** 33,979.
 - Vocales: 300 por cada una. 1500 en total.
 - Consonantes: [Z]: 600, [b]: 2500, [d]: 2500, [f]: 1500, [g]: 1600, [k]: 2500, [l]: 2500, [m]: 2500, [n]: 2500, [n~]: 320, [p]: 2500, [r]: 1500, [r(): 2500, [s]: 2500, [t]: 2500, [tS]: 359, [x]: 1600. 32,479 en total.

- **Nodo Padre 1:** Se asignó una sobrecarga de datos para los datasets de los alófonos vocales ([a], [e], [i], [o] y [u]) igual a 4,230 y una cantidad de datos igual a 300 por cada alófono restante. Por ejemplo, el alófono vocal [a] cuenta con una cantidad de 4,230 de datos y los demás alófonos vocales con una cantidad de 300 datos respectivamente, dando así un total de datos de 5,430 en el dataset.

- **Nodo Padre 2**
 - **DataSet del alófono [Z]:** 648 en total. Sobrecarga de datos para el alófono [Z] igual a 600. Dejando a las demás consonantes con 3 cada uno (48 en total).
 - **DataSets alófonos [b], [d], [k], [l], [m], [n], [p], [r(), [s] y [t]:** 3,300 en total. Al igual que en el Nodo Padre 1 se asignó una sobrecarga de datos de 2,500 dejando por consecuencia al resto con una cantidad de 50 cada uno (800 en total).
 - **DataSet de los alófonos [f] y [r]:** 1,820 en total. Al igual que en el Nodo Padre 1 se asignó una sobrecarga de datos de 1,500 dejando por consecuencia al resto con una cantidad de 20 cada uno (320 en total).
 - **DataSet de los alófonos [g] y [x]:** 1,920 en total. Al igual que en el Nodo Padre 1 se asignó una sobrecarga de datos de 1,600 dejando por consecuencia al resto con una cantidad de 20 cada uno (320 en total).
 - **Dataset del alófono [n~]:** 352 en total. Sobrecarga de datos para el alófono [n~] igual a 320. Dejando a las demás consonantes con 2 cada uno (32 en total).
 - **Dataset del alófono [tS]:** 391 en total. Sobrecarga de datos para el alófono [tS] igual a 359. Dejando a las demás consonantes con 2 cada uno (32 en total).